**Zadania przygtowawcze 2**

*Piotr Chrz ˛astowski-Wachtel*

1. Napisz funkcj˛e

bool Iloczyny6(int T[], int n, int k),

która przyjmie warto´s´c "true", wtedy i tylko wtedy gdy iloczyn liczb zapisanych w kazdym ˙ segmencie tablicy T długo´sci k jest podzielny przez 6 dla k,n>0.

2. Tablica A zawiera albo liczby dodatnie, albo warto´sci -1, przy czym wszystkie warto´sci -1 wyst˛epuj ˛a w jednym spójnym segmencie. Ponadto dla dowolnych *i*, *j* takich ze 0 ˙ *≤ i ≤ j <*n zachodzi (*A*[*i*] *>* 0 *∧ A*[*j*] *>* 0) *→ A*[*i*] *≤ A*[*j*]. Napisz funkcj˛e int znajdz(int A[], int n, int x) , która znajdzie w tablicy A indeks zadanej liczby dodatniej x, o ile ona si˛e tam znajduje. Je´sli liczby x nie ma w A, to warto´sci ˛a funkcji powinno by´c -1.

3. Napisz funkcj˛e bool takiesame(int A[], int N), która sprawdzi, czy z zapisanego w tablicy ci ˛agu 2N liczb całkowitych mozna wybra´c N-elementowy podci ˛ag tak, by ci ˛agi ˙ elementów wybranych i niewybranych były takie same.

4. Dla tablicy *A* oraz liczby całkowitej *k* powiemy, ze segment ˙ A[d..g] jest *k*-płaski, je´sli dla kazdych ˙ *i*, *j* takich, ze˙ *d ≤ i*, *j ≤ g* zachodzi nierówno´s´c *A*[*i*] *− A*[*j*] *≤ k*. Napisz funkcj˛e int kaplaski(int A[], int n, int k), która dla tablicy *A* długo´sci *n* wyz naczy długo´s´c najdłuzszego ˙ *k*-płaskiego segmentu.

5. Niech *T =* {0,...,*n −*1}. Funkcja *f* : *T → T* jest niemalej ˛aca, a jej warto´sci s ˛a zapisane w tablicy int F[n] tak, ze˙ *f* (*k*) *= F*[*k*] dla *k =* 0,...,*n−*1. Definiujemy funkcj˛e *g f* : *T → T* w nast˛epuj ˛acy sposób: *g f* (*k*) *= |*{*a ∈ T* : *f* (*a*) *= k*}*|*, gdzie przez *|X|* oznaczamy liczno´s´c (moc) zbioru *X*. Napisz fragment programu, który zmiennej *rekord* nada warto´s´c równ ˛a maksimum spo´sród warto´sci przeciwobrazów funkcji *g f* z jednoelementowych podzbiorów

zbioru T, czyli *max*(*|*{*g f −*1({*k*})*|* : *k =* 1,...,*n*}).

6. Przez segment słowa *v ∈ A∗*składaj ˛acy si˛e z samych liter *x* nazwiemy takie słowo *w = xk* dla pewnego *x ∈ A* oraz *k ∈ N+* , ze˙ *v = w0ww00* dla pewnych *w0*,*w00 ∈ A∗*. Seg ment taki jest maksymalny, je´sli ani *w0* nie ko ´nczy si˛e na *x*, ani *w00* nie zaczyna si˛e od *x*. Napisz gramatyk˛e generuj ˛ac ˛a wszystkie słowa *v* nad alfabetem {*a*,*b*,*c*} o tej wła´sci wo´sci, ze ka ˙ zdy maksymalny segment zło ˙ zony z samych liter ˙ *b* w słowie *v* musi by´c bezpo´srednio poprzedzony przez dłuzszy segment zło ˙ zony z samych liter ˙ *a*.

7. Dane s ˛a dwie funkcje *f* , *g* : {0, 1}*∗ → Z* okre´slone w nast˛epuj ˛acy sposób: *f* (*ε*) *=* 0, *f* (0*w*) *= g* (*w*) *+*1, *f* (1*w*) *= g* (*w*) *−*1, *g* (*ε*) *=* 0, *g* (0*w*) *= f* (*w*) *−*1, *g* (1*w*) *= f* (*w*) *+*1 dla *w ∈* {0, 1}*∗*. Zdefiniujmy j˛ezyk *L =* {*w ∈* {0, 1}*∗*: *f* (*w*) *=* 0}. Napisz funkcj˛e bool JestL(int T[], int n), która sprawdzi, czy słowo zapisane w tablicy *T* nalezy do j˛ezyka ˙ *L*.

8. Dana jest nast˛epuj ˛aca funkcja:

1

int coto(int n)

{ int i,x,y;

x=0; y=1; i=2;

while (i<=n)

{ x+=y;

y=x-y;

i+=2;

}

if (i%2) return y;

else return x;

}

Okre´sl dla kazdego naturalnego ˙ *n*, co jest warto´sci ˛a funkcji coto(n) i udowodnij to. 9. Dana jest nast˛epuj ˛aca funkcja:

int coto2(int n)

{ int x,y;

x=0; y=1;

while (y<=n)

{ x-=y;

y-=x;

}

if (i%2) return y;

else return x;

}

Okre´sl dla kazdego naturalnego ˙ *n*, co jest warto´sci ˛a funkcji coto(n) i udowodnij to.

10. Dany jest ci ˛ag prostok ˛atów *P*1,...,*Pn*. Tablice int A[],B[] długo´sci *n >* 0 reprezentuj ˛a długo´sci boków tych prostok ˛atów tak, ze dla ka ˙ zdego ˙ *i =* 0,...,*n −*1 para (A[i],B[i]) okre´sla długo´sci boków prostok ˛ata *Pi+*1. Ci ˛ag tych prostok ˛atów ma t˛e szczególn ˛a włas no´s´c, ze tablica ˙ A jest posortowana niemalej ˛aco, a tablica B nierosn ˛aco. Napisz funkcj˛e float int minprzek(int A[], int B[], int n), która wyznaczy indeks prostok ˛ata

o najkrótszej przek ˛atnej.

11. Napisz funkcj˛e float mindist(float A[], float B[], int n), która obliczy warto´s´c *min*(*|A*[*k*]*−B*[*k*]*|* : 0 *≤ k < n*), przy załozeniu, ˙ ze tablica ˙ A jest posortowana rosn ˛aco, a B malej ˛aco.

12. Podaj gramatyk˛e j˛ezyka *L =* {*w ∈* {*a*,*b*}*∗*: #(*a*,*w*) *=* 2#(*b*,*w*)}.

13. W tablicy float A[] długo´sci n jest zapisana wysoko´s´c pokrywy ´snieznej na dachu ˙ pewnego budynku w dniach 1..n i wiemy, ze jest to ci ˛ag słabo bitoniczny (najpierw ˙ niemalej ˛acy, potem nierosn ˛acy). Kazdy centymetr ´swie ˙ zego ´sniegu (z danego dnia) wa ˙ zy˙

2

nw, za´s centymetr ´sniegu z dnia poprzedniego jest c razy ci˛ezszy ni ˙ z w dniu poprzed- ˙ nim. Kolejne warstwy ´sniegu układaj ˛a si˛e na poprzednich; je´sli ´snieg topnieje, to dzieje sie to w kolejno´sci odwrotnej, czyli najpierw topnieje ´snieg z wczoraj, potem zprzed wczoraj itd. Napisz funkcj˛e bool zaciezko(int A[], float c, float r, float nw), która sprawdzi, czy był taki dzie ´n, w którym ci˛ezar ´sniegu na dachu przekroczył ˙ warto´s´c krytyczn ˛a r.

14. Niech A b˛edzie tablic ˛a liczb całkowitych o dodatniej parzystej dłogo´sci. Napisz funkcj˛e bool kantybiton(int A[] int n, sprawdzaj ˛ac ˛a, czy istnieje taka liczba *k* z prze działu 0,...,*n −* 2, ze tablica ˙ *B*, równa tablicy *A* przesuni˛etej cyklicznie o k elementów w prawo, jest antybitoniczna, czyli od pocz ˛atku do pewnego miejsca malej ˛aca i od tego samego miejsca do ko ´nca rosn ˛aca.

15. Rozwazamy funkcj˛e ˙ *f* : {0,...,*n−*1} *→* {0,...,*n−*1}. Dla kazdego dodatniego ˙ *k* okre´slamy *fk*(*x*) *= f* (*x*) dla k=1 oraz *fk*(*x*) *= fk−*1(*f* (*x*)) dla *k >* 1. Powiemy, ze˙ *y* jest zalezne od ˙ *x* je´sli istnieje *k >* 0 takie, ze˙ *y = fk*(*x*). O funkcji *f* wiemy tyle, ze wyznaczona przez ˙ ni ˛a relacja zalezno´sci jest antysymetryczna. Dwa elementy ˙ *x* i *y* s ˛a niezalezne, je´sli ani ˙ *x* nie jest zalezne od ˙ *y* ani *y* od *x*. Zbiór *X* nazwiemy niezaleznym, je´sli ka ˙ zde dwa ˙ jego elementy s ˛a niezalezne. Funkcj˛e ˙ *f* definiujemy za pomoc ˛a tablicy długo´sci *n* o warto´sciach ze zbioru 0,...,*n −*1. Napisz funkcj˛e int maxniezalezne(int f[], int n), która wyznaczy liczno´s´c najliczniejszego niezaleznego podzbioru zbioru {0,..., ˙ *n−*1} wyznaczonego przez funkcj˛e *f* .

16. Tablica *A* zawiera same jedynki i zera. Napisz funkcj˛e int maxrot(int A[], int n), która wyznaczy takie *k*, zeby jedynki i zera w tablicy powstałej przez przesuni˛ecie ˙ *A* o *k* pozycji w prawo ułozyły si˛e w mo ˙ zliwie du ˙ z ˛a liczb˛e w zapisie dwójkowym. ˙ *Zrobienie tegow czasie liniowym jest mozliwe, ale wykracza poza zakres materiału tego semestru ˙*

17. Dane s ˛a dwie liczby rzeczywiste. Kazda z nich jest reprezentowana w dwóch zeroje- ˙ dynkowych tablicach: w tablicy n-elementowej zapisana jest cecha, a w m-elementowej mantysa. Napisz funkcj˛e bool wieksza(int C1[], int M1[], int C2[], int M2[], int n, int m), która sprawdzi, czy liczba zapisana w systemie zmiennopozycyjnym w kodzie uzupełnieniowym o cesze C1 i mantysie M1 jest wi˛eksza niz liczba o cesze C2 i ˙ mantysie M2. Zakładamy, ze minimalna cecha reprezentuje 0 i ˙ ze nie ma ukrytego bitu ˙ jednej drugiej.

18. Dla typów, jak w zadaniu poprzednim, wygeneruj kolejn ˛a co do wielko´sci liczb˛e rzeczy wist ˛a do danej. int kolejna(int C[], int M[], int n, int m). Je´sli mieli´smy na jwi˛eksz ˛a mozliw ˛a liczb˛e w tablicach C i M, to zostaw tablice w spokoju i niczego nie ˙ zmieniaj.

19. Oblicz bł ˛ad wzgl˛edny wykonania działania 27*+*311 w systemie zmiennopozycyjnym z wykładu.

3